



TU Clausthal
Clausthal University of Technology

Supervised Median Clustering

**Barbara Hammer¹, Alexander Hasenfuss¹,
Frank-Michael Schleif², and Thomas Villmann³**

IfI Technical Report Series

IfI-06-09



Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany

Editor of the series: Jürgen Dix

Technical editor: Wojciech Jamroga

Contact: wjamroga@in.tu-clausthal.de

URL: <http://www.in.tu-clausthal.de/forschung/technical-reports/>

ISSN: 1860-8477

The IfI Review Board

Prof. Dr. Jürgen Dix (Theoretical Computer Science/Computational Intelligence)

Prof. Dr. Klaus Ecker (Applied Computer Science)

Prof. Dr. Barbara Hammer (Theoretical Foundations of Computer Science)

Prof. Dr. Kai Hormann (Computer Graphics)

Prof. Dr. Gerhard R. Joubert (Practical Computer Science)

Prof. Dr. Ingbert Kupka (Theoretical Computer Science)

Prof. Dr. Wilfried Lex (Mathematical Foundations of Computer Science)

Prof. Dr. Jörg Müller (Agent Systems)

Dr. Frank Padberg (Software Engineering)

Prof. Dr.-Ing. Dr. rer. nat. habil. Harald Richter (Technical Computer Science)

Prof. Dr. Gabriel Zachmann (Computer Graphics)

Supervised Median Clustering

Barbara Hammer¹, Alexander Hasenfuss¹, Frank-Michael Schleif², and Thomas Villmann³

1 - Clausthal University of Technology, Institute of Computer Science, Clausthal-Zellerfeld,
Germany

2 - University of Leipzig, Institute of Computer Science, Germany

3 - University of Leipzig, Clinic for Psychotherapy, Leipzig, Germany

Abstract

Prototype based clustering and classification algorithms constitute very intuitive and powerful machine learning tools for a variety of application areas. They combine simple training algorithms and easy interpretability by means of prototype inspection. However, the classical methods are restricted to data embedded in a real vector space and thus, have only limited applicability to complex data as occurs in bioinformatics or symbolic areas. Recently, extensions of unsupervised prototype based clustering to proximity data, i.e. data characterized in terms of a distance matrix only, have been proposed. Since the distance matrix constitutes a universal interface, this opens the way towards an application of efficient prototype based methods for general data. In this contribution, we transfer this idea to supervised scenarios, proposing a prototype based classification method for general proximity data.

1 Introduction

Prototype-based classification such as learning vector quantization (LVQ) [9] constitutes an intuitive machine learning technique which represents classes by typical prototype locations and assigns labels to new data points by means of a winner-takes-all rule. This principle is particularly striking because of its simple learning rule, its intuitive way to deal with several classes, and its easy interpretability. Unlike feedforward networks or support vector machines (SVM), the method provides insight into the classification behavior by an inspection of the prototypes: these are located in the data space and thus constitute prototypical class representatives. Interestingly, the generalization behavior of prototype-based techniques is quite robust: large margin generalization bounds can be derived which only depend on the hypothesis margin but not on the number of parameters of the model, similar to SVMs [3].

Original LVQ, however, has been proposed for vectorial data only, such that its applicability to complex domains is limited. There exist powerful extensions to incorporate general kernels [7, 13], however, these methods require a Hilbert space of the kernel

where data are embedded. In addition, [7] assumes differentiability of the kernel and cannot easily be applied to discrete data. The approach [13] extends solutions in terms of the training data and does not yield sparse solutions in terms of prototypical locations. Here, we are interested in extensions of prototype-based classification to general proximity data which is given in terms of a general distance metric. Thereby, the metric need not be given in explicit terms, nor need it be positive semidefinite such that no underlying kernel can be identified. Such data occur frequently in fields such as psychology, Neuroscience, molecular biology, or economics [5].

Various approaches to deal with general proximity data have been proposed in the literature, e.g. unsupervised clustering algorithms [14, 16] and several supervised large margin methods which enlarge SVM or borrow ideas from SVM, respectively [4, 8, 12]. These methods, however, represent solutions in terms of support vectors or similar quantities instead of typical locations of the data such as prototype based methods. In addition, the training algorithms usually rely on a linear or quadratic programming problem which is derived via a primal-dual problem formulation, thus training is less intuitive compared to prototype-based methods. Simple k-nearest neighbor methods constitute another alternative which can be combined with arbitrary distance matrices. However, solutions store all data points and are not sparse. Here, we directly extend ideas from prototype-based learning to general proximity data, preserving easy interpretability and sparsity of the models.

The main technique to achieve this goal is borrowed from unsupervised clustering algorithms: batch variants of popular algorithms such as the self-organizing map (SOM), neural gas (NG), and k-means can be transferred to general proximity data by means of the generalized median [2, 10]. This restricts valid prototype locations to the given data points and, therefore, does not use a surrounding vector space of the data. We combine this method with supervised batch neural gas, which has been recently proposed [6]. Convergence of the algorithm is guaranteed, and we demonstrate its applicability in several examples in this article.

2 Supervised neural gas

Neural gas (NG) as presented in [11] constitutes a very robust unsupervised clustering algorithm. Assume data vectors $\mathbf{v} \in \mathbb{R}^d$ are given as stimuli, distributed according to an underlying probability distribution $P(\mathbf{v})$. The aim of prototype-based unsupervised clustering is to find a number of prototypes or weight vectors $\mathbf{w}_i \in \mathbb{R}^d$, $i = 1, \dots, n$ representing the data points faithfully, e.g. measured in terms of the average deviation of a data point from its respective closest prototype. The objective of NG is a minimization of the cost function

$$E_{\text{NG}}(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^n \int h_{\lambda}(k_i(\mathbf{v}, W)) \cdot (\mathbf{v} - \mathbf{w}_i)^2 P(\mathbf{v}) d\mathbf{v}$$

where $k_i(\mathbf{v}, W) = |\{\mathbf{w}_j \mid (\mathbf{v} - \mathbf{w}_j)^2 < (\mathbf{v} - \mathbf{w}_i)^2\}|$ is the rank of prototype i , $h_\lambda(t)$ is a Gaussian shaped curve such as $h_\lambda(t) = \exp(-t/\lambda)$ with neighborhood range $\lambda > 0$, and $C(\lambda)$ is a normalization constant. Typically, online adaptation takes place by means of a stochastic gradient descent method. The resulting learning rule adapts all prototypes after the presentation of each stimulus by a small step, whereby the rank determines the adaptation strength. Recently, an alternative batch adaptation scheme for this cost function has been proposed which, for a given finite training set, in turn, determines the rank $k_i(\mathbf{v}, W)$ according to fixed prototype locations and the prototype locations as average of all training points weighted according to their rank, until convergence. Batch adaptation can be interpreted as Newton optimization of the cost function, and often a fast convergence can be observed compared to online adaptation.

Batch adaptation provides an interface towards clustering general proximity data. In this case, only pairwise distances of the data points are given but in general no embedding within a real-vector space is available. The euclidian distance is substituted by the given proximities, and optimization of prototypes takes place within the discrete space given by the data points, as proposed in [2].

For supervised classification, additional information in the form of class labels is available. That means, labels y_i are given for every data point \mathbf{v}_i . We assume that $y_i \in \mathbb{R}^d$, d being the number of classes. This notion subsumes crisp classification with unary encoded class information as well as fuzzy assignments. Obviously, this information can be incorporated into standard NG as well as median NG by means of posterior labeling. I.e. the average of the labels of all training patterns in its respective receptive field are assigned to a prototype. We refer to these methods as BNG (batch neural gas) and BNGMedian. However, posterior labeling has the drawback that prototype locations are determined only based on the input data and labels are not taken into account during the training process.

As an alternative, the additional class information can be incorporated into the learning process by an extension of the overall cost function. First promising steps into this direction can be found in the approach [15] for online NG, however, the method proposed in [15] cannot be transferred to median versions. A batch variant, supervised batch NG (SBNG), recently proposed in [6], is based on the cost

$$\begin{aligned} E_{\text{SBNG}}(W, Y) = & (1 - \alpha) \cdot \frac{1}{2C(\lambda)} \sum_{i=1}^n \int h_\lambda(k_i(\mathbf{v}, y, W, Y)) \cdot (\mathbf{v} - \mathbf{w}_i)^2 P(\mathbf{v}) d\mathbf{v} \\ & + \alpha \cdot \frac{1}{2C(\lambda)} \sum_{i=1}^n \int h_\lambda(k_i(\mathbf{v}, y, W, Y)) \cdot (y - Y_i)^2 P(\mathbf{v}) d\mathbf{v} \end{aligned}$$

where $k_i(\mathbf{v}, y, W, Y) = |\{\mathbf{w}_j \mid (1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_j)^2 + \alpha \cdot (y - Y_j)^2 < (1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_i)^2 + \alpha \cdot (y - Y_i)^2\}|$ denotes the rank of prototype i measured according to the closeness of the current data point and the prototype weight and labeling. $\alpha \in [0, 1]$ constitutes a weighting of the two objectives, label learning and a distribution of prototypes among the data. Thereby, each prototype \mathbf{w}_i is equipped with an additional

vector $Y_i \in \mathbb{R}^d$ which should represent the class labels of data points in the receptive field as accurately as possible and which is automatically adapted during training. In particular, it is possible to learn appropriate crisp or fuzzy labels of the prototypes.

Batch adaptation schemes suppose that a finite set of training data $(\mathbf{v}_1, y_1), \dots, (\mathbf{v}_p, y_p)$ is given in advance. For this finite training set, batch optimization determines in turn the hidden variables $k_{ij} := k_i(\mathbf{v}_j, y_j, W, Y)$ and the weights and labels W and Y until convergence. This yields the following update rules of SBNG

- (1) For given W, Y , set $k_{ij} = |\{\mathbf{w}_l \mid (1 - \alpha) \cdot (\mathbf{v}_j - \mathbf{w}_l)^2 + \alpha \cdot (y_j - Y_l)^2 \leq (1 - \alpha) \cdot (\mathbf{v}_j - \mathbf{w}_i)^2 + \alpha \cdot (y_j - Y_i)^2\}|$ as the rank of prototype i given \mathbf{v}_j .
- (2) For fixed k_{ij} , set $\mathbf{w}_i = \sum_j h_\lambda(k_{ij}) \cdot \mathbf{v}_j / \sum_j h_\lambda(k_{ij})$, and $Y_i = \sum_j h_\lambda(k_{ij}) \cdot y_j / \sum_j h_\lambda(k_{ij})$.

Note that the assignments of the receptive fields and the rank depend on the closeness of the prototype as well as the correctness of its class label. This has the effect that the prototypes of SBNG better account for cluster borders of labeled data points, whereas NG only follows the overall statistics. It has been shown in [6] that this scheme converges in a finite number of steps towards a local optimum of the cost term under mild conditions on the output.

3 Supervised median neural gas

Assume data are not embedded in a euclidian vector space, instead, pairwise proximities $d_{ij} = d(\mathbf{v}_i, \mathbf{v}_j)$ are available. Thereby, there are no assumptions on these values such as symmetry or positive definiteness. In this case, the prototype locations cannot be chosen arbitrarily but discrete adaptation has to take place. We assume that prototypes are located in the data space, i.e. $w_i = \mathbf{v}_j$ for some j . We can transfer the cost function of SBNG to these data, yielding the cost function of supervised median neural gas (SBNGMedian) for finite data

$$\begin{aligned} \hat{E}_{\text{SBNGMedian}}(W, Y) &= (1 - \alpha) \cdot \frac{1}{2C(\lambda)} \sum_{i=1}^n \sum_{j=1}^p h_\lambda(k_i(\mathbf{v}_j, y_j, W, Y)) \cdot d_{il_j}^2 \\ &\quad + \alpha \cdot \frac{1}{2C(\lambda)} \sum_{i=1}^n \sum_{j=1}^p h_\lambda(k_i(\mathbf{v}_j, y_j, W, Y)) \cdot (y_j - Y_i)^2 \end{aligned}$$

where $\mathbf{w}_j = \mathbf{v}_{l_j}$ and $k_i(\mathbf{v}_j, y_j, W, Y) = |\{\mathbf{w}_k \mid (1 - \alpha) \cdot d_{il_k}^2 + \alpha(y_j - Y_k)^2 < (1 - \alpha) \cdot d_{il_j}^2 + \alpha(y_j - Y_i)^2\}|$. For batch optimization, we optimize in turn hidden variables $k_{ij} := k_i(\mathbf{v}_j, y_j, W, Y)$ for fixed weights and labels with the constraint, that k_{ij} yields a permutation of $\{0, \dots, n - 1\}$ for every j ; and weights and labels for fixed hidden variables k_{ij} . Thereby, no embedding vector space exists for the weights, such that we have to restrict the optimization to the set of input vectors. Weights are set to the

so-called generalized median, i.e. the respective location in the (finite, discrete) input space which optimizes the cost function for fixed hidden variables. Thus, supervised median neural gas is given by the formulas.

- (1) For given W, Y , set $k_{ij} = |\{\mathbf{w}_k \mid (1 - \alpha) \cdot d_{l_{kj}}^2 + \alpha \cdot (y_j - Y_i)^2 \leq (1 - \alpha) \cdot d_{l_{ij}}^2 + \alpha \cdot (y_j - Y_i)^2\}|$ as the rank of prototype i given \mathbf{v}_j .
- (2) For fixed k_{ij} , set $\mathbf{w}_i = \mathbf{v}_l$ for which $\sum_{i,j} h_\lambda(k_{ij}) \cdot d_{l_{ij}}^2$ is minimum and $Y_i = \sum_j h_\lambda(k_{ij}) \cdot y_j / \sum_j h_\lambda(k_{ij})$.

The minimum in step (2) is determined by extensive search, such that the complexity of one step is of order p^2 . This way, the prototype locations are adapted within the space of input features according to the given statistical information incorporating label information through the ranks. Thereby, the prototype labels are adapted automatically, enabling an automatic optimization of the number of prototypes representing a class as well as a fuzzy classification. For crisp classification, a further improvement is possible if the labels are not yet exactly settled in optima of the cost function: we can posteriorly label the prototypes according to the receptive fields by majority vote. This yields an optimum assignment for the labels once the receptive fields (which incorporate label information) are fixed. We refer to this method by SBNGMedian⁺. Depending on whether SBNGMedian did already converge, this yields a further improvement of the method. Note that it can be shown in the same way as presented in [6] that the SBNGMedian algorithm converges in a finite number of adaptation steps.

4 Experiments

We test the problem in several experiments, including the case of real-valued data and the standard euclidian norm where a direct comparison to SBNG is possible as well as proximity data for which SBNG cannot be applied.

Wisconsin Breast Cancer Database

The Wisconsin breast cancer data consists of nearly 600 data points described by 30 real-valued input features which are to be separated into 2 classes. We train 40 neurons using 200 epochs. The dataset is randomly split, and the result on the test set averaged over 250 runs is reported. Since data are contained in the euclidian space, standard SBNG and BNG can be applied as well, as reported in [6]. These versions yield an accuracy of 94.11% (BNG) and 95.45% (SBNG for mixing parameter $\alpha = 0.9$), respectively. The results of median versions yield 93.3% (BNGMedian) and 94.26% (SBNGMedian) for the optimum parameter $\alpha = 0.6\%$, respectively. Thus, the classification accuracy is slightly improved (about 1%) in both cases by an extension of the cost function of BNG to label information. The accuracy of the median versions approaches the accuracy of the euclidian variants, whereby about 1% accuracy is lost due

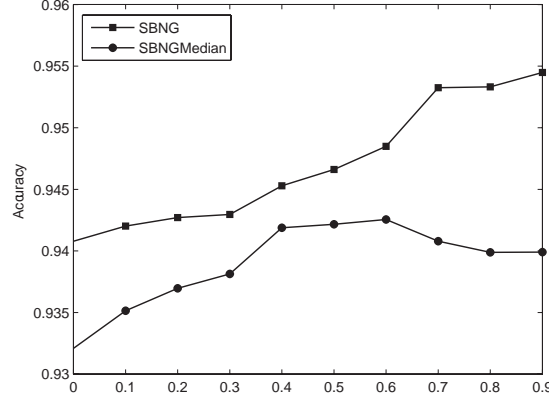


Figure 1: Results of supervised median neural gas with and without posterior labeling in comparison to standard median neural gas for different values of the mixing parameter α ranging from 0 to 0.9 on the wisconsin breast cancer dataset.

to the fact that the prototype locations are restricted to the discrete locations provided by the input data. The influence of the mixing parameter α on the classification accuracy is depicted for both, the standard batch version and the median version in Fig. 1. $\alpha = 0$ corresponds to fully unsupervised training. As can be seen, an integration of label information i.e. $\alpha > 0$ allows an improvement of the classification accuracy whereby the optimum value depends on the situation at hand.

Chicken Pieces Silhouettes Database

The task is to classify 446 silhouettes of chicken pieces into 5 categories (wing, back, drumstick, thigh and back, breast). As reported in [12], data are preprocessed by representing each silhouette as a string of the angles of consecutive tangential line pieces, whereby rotation and scale invariance is included. Thereby, the length of tangents is 20. Strings are compared using the edit distance, whereby insertions/deletions cost 60, and the angle difference is taken otherwise. Thus in this case, discrete patterns are dealt with by means of a proximity matrix. We train median clustering on these data using 40 neurons per run and 500 epochs. In each, the dataset is randomly split in a training and test set and the average classification accuracy of the test set for 100 runs is reported in Fig. 2 for mixture values α ranging from 0 to 0.9. The choice $\alpha = 0$ yields standard unsupervised median neural gas. Obviously, for this data set, larger values of α allow a better classification reaching an accuracy above 0.83 for values close to 1, i.e. more than 10% improvement compared to an unsupervised version. Thereby, posterior label-

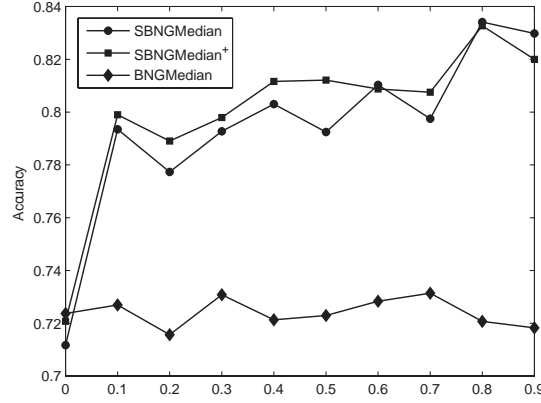


Figure 2: Results of supervised median neural gas with and without posterior labeling in comparison to standard median neural gas for different values of the mixing parameter α ranging from 0 to 0.9 on the chicken pieces dataset.

ing does slightly improve SBNGMedian. Interestingly, these results are better than the results reported in [12] for k-NN classification (0.74) and an SVM specifically adapted to proximity data given by the edit distance (0.81).

Chromosomes

The Copenhagen chromosomes database consists of 4200 data points of grey level images of chromosomes as described in [12]. The silhouettes are represented as strings corresponding to the thickness of the gray level and compared using the edit distance as before [12]. The algorithms have been run using 100 neurons and 100 epochs per run. The result for different mixing parameters α can be seen in Fig. 3. The reported results consist of the test set accuracy averaged over 10 runs. Again, integration of class labels into training allows an improvement of about 3%. The optimum value 0.87 is achieved for α close to 1. In [12], a better accuracy of 0.91 is reported for k-NN and SVM, however, this is achieved only on a subset of the data consisting of less than 1000 points.

Proteins

The task is to classify 226 samples of proteins into 4 classes, whereby an evolutionary distance between the points is taken as proximity as described in [8]. We used 30 neurons and 300 epochs per run. The accuracy on the test set averaged over 100

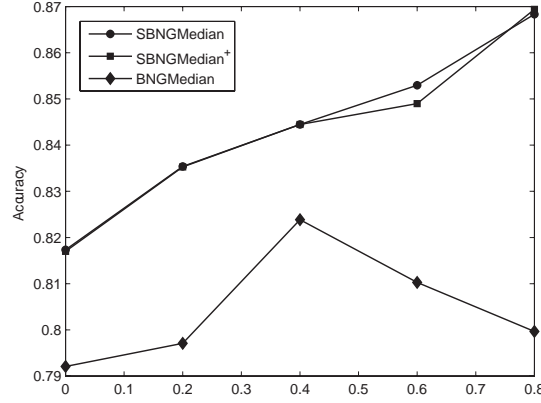


Figure 3: Results of the methods for the chromosomes database and varying mixing parameter α .

runs is reported for different values α in Fig. 4. Here, the optimum can be observed for $\alpha = 0.2$, thus indicating that the statistics of the inputs guides the way towards a good classification accuracy. However, an integration of the labels with small mixing parameter improves the accuracy by nearly 10% compared to the fully unsupervised SBNGMedian. Unlike the results reported in [8] for SVM which use one-versus-rest encoding, the classification of SBNGMedian and SBNG is given by one single classifier.

5 Conclusions

We have presented an extension of well established clustering methods to a variant which can deal with supervised classification of proximity data by means of a prototype based model. This model benefits from the strength of prototype based methods such as LVQ: simple interpretability of the models, sparse representation by a priorsly fixed number of prototypes, and intuitive training. Standard LVQ is extended to deal with adaptive and possibly fuzzy labeling and general proximity data. In all cases, integration of class information during training significantly improves the accuracy, partially by up to 10%. Thereby, the method is even better than SVM as reported in [12] for the chicken pieces dataset.

So far, the determination of the generalized median has been done by extensive search such that the complexity of one epoch is quadratic in the number of training patterns. The approaches [10, 1]. present improved variants which compute the median

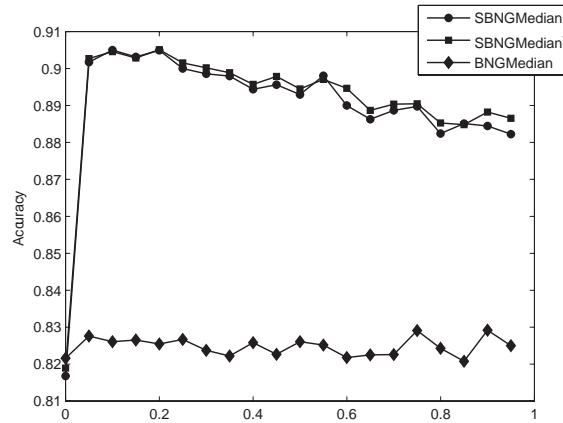


Figure 4: Results of the methods for the protein database and varying mixing parameter α .

based on approximations e.g. reducing to an appropriate subset of points. The transfer of these methods will be the subject of future work.

References

- [1] B. Conan-Guez, F. Rossi, and A. El Golli (2005), A fast algorithm for the self-organizing map on dissimilarity data, in *Workshop on Self-Organizing Maps*, 561-568.
- [2] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann (2006), Batch and median neural gas, *Neural Networks*, **19**:762-771.
- [3] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby (2002), Margin analysis of the LVQ algorithm, *NIPS'2002*.
- [4] T. Graepel, R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K. Robert-Müller, K. Obermayer, and R. Williamson (1999), Classification on Proximity Data with LP-Machines. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, pages 304-309.
- [5] T. Graepel and K. Obermayer (1999), A stochastic self-organizing map for proximity data, *Neural Computation* **11**:139-155.

References

- [6] B. Hammer, A. Hasenfuss, F.-M. Schleif, and T. Villmann (2006), Supervised batch neural gas, In F. Schwenker, and S. Marinai (Eds.), *ANNPR 2006, Springer Lecture Notes in Artificial Intelligence* **4087**:33-45.
- [7] B. Hammer, M. Strickert, and T. Villmann (2005), Supervised neural gas with general similarity measure, *Neural Processing Letters* **21**(1), 21-44.
- [8] B. Haasdonk and C. Bahlmann (2004), Learning with distance substitution kernels, in *Pattern Recognition - Proc. of the 26th DAGM Symposium*.
- [9] T. Kohonen (1995), *Self-Organizing Maps*, Springer.
- [10] T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for nonvectorial data, *Neural Networks* **15**:945-952.
- [11] T. Martinetz, S.G. Berkovich, and K.J. Schulten (1993), 'Neural-gas' network for vector quantization and its application to time-series prediction, *IEEE Transactions on Neural Networks* **4**:558-569.
- [12] M. Neuhaus and H. Bunke (2006), Edit distance based kernel functions for structural pattern classification, to appear in *Pattern Recognition*.
- [13] A.K. Qin and P.N. Suganthan (2004), A Novel Kernel Prototype-Based Learning Algorithm, *17th Int. Conf. on Pattern Recognition, ICPR'04*.
- [14] S. Seo and K. Obermayer (2004), Self-organizing maps and clustering methods for matrix data, *Neural Networks* **17**:1211-1230.
- [15] T. Villmann, B. Hammer, F. Schleif, T. Geweniger, and W. Herrmann (2006), Fuzzy classification by fuzzy labeled neural gas, *Neural Networks*, accepted.
- [16] S. Zhong and J. Ghosh (2003), A unified framework for model-based clustering, *Journal of Machine Learning Research* **4**:1001-1037.